**Aplicatia mea pentru SO**

**Mediu rulare Linux:**

- Oracle VM VirtualBox

**Configurare Linux:**

sudo apt-get update

sudo apt-get upgrade

sudo apt-get install openssh-server openssh-client

service ssh start

service ssh status

sudo apt-get install virtualbox-guest-dkms

sudo mount -t vboxsf vbShare mntShare

sudo apt-get install apache2

sudo apache2ctl configtest

sudo ufw allow in "Apache Full"

sudo apt-get install mysql-server

mysql\_secure\_installation

sudo apt-get install php libapache2-mod-php php-mcrypt php-mysql

sudo systemctl restart apache2

sudo systemctl status apache2

sudo apt-get install libssh2-1 php-ssh2 -y

<https://www.cyberciti.biz/faq/linux-unix-running-sudo-command-without-a-password/>

**Configurare user MySQL server**

https://stackoverflow.com/questions/39281594/error-1698-28000-access-denied-for-user-rootlocalhost

$ sudo mysql -u root # I had to use "sudo" since is new installation

mysql> USE mysql;

mysql> CREATE USER 'YOUR\_SYSTEM\_USER'@’localhost IDENTIFIED BY 'YOUR\_PASSWORD';

mysql> GRANT ALL PRIVILEGES ON \*.\* TO 'YOUR\_SYSTEM\_USER'@'localhost';

mysql> UPDATE user SET plugin='mysql\_native\_password' WHERE User='YOUR\_SYSTEM\_USER';

mysql> FLUSH PRIVILEGES;

mysql> exit;

$ service mysql restart

**Limit execution time**

sleep.sh:

#!/bin/sh

sleep 100

$ timeout 1 ./sleep.sh -> makes the script (and any command) run for x seconds

**Server pentru testare**

- imi voi configura eu prorpiul server cu care voi comunica prin SSH (fie utilizator va avea un cont de utilizator pe serverul Linux)

- fenrir

**Server pentru aplicatia web**

- Apache Linux

Nu am putut adauga libraria pentru SSH2 pe Windows.

- XAMPP Windows/Linux

**Limbaje**

- front-end: HTML+CSS

- back-end: PHP

**Autentificare**

- voi folosi autentificarea prin SSH la Fenrir

- resource **ssh2\_connect** ( string $host [, int $port = 22 [, array $methods [, array $callbacks ]]] )

**Capitole**

- comenzi bash

- scripturi

- programe C Linux

**Evaluarea**

- comenzi bash

- voi stoca comanda celui care adauga problema

- cand cineva ii rezolva problema, prin SSH ma voi conecta la fenrir (cu contul celui care rezolva problema) si voi rula pe rand solutia buna a problemei si solutia celui ce rezolva problema si voi compara rezultatle

- resource **ssh2\_exec** ( resource $session , string $command [, string $pty [, array $env [, int $width = 80 [, int $height = 25 [, int $width\_height\_type = SSH2\_TERM\_UNIT\_CHARS ]]]]] )

- scripturi

- voi stoca codul scriptului celui care adauga problema

- cand cineva ii rezolva problema, prin SSH voi trimite scriptul pe fenrir (cu contul celui care rezolva problema) si il voi executa

- apoi voi trimite si voi executa solutia corecta a problemei

- voi compara cele 2 rezultate

- citirea si afisarea se vor face cu fisiere

- bool **ssh2\_scp\_send** ( resource $session , string $local\_file , string $remote\_file [, int $create\_mode = 0644 ] )

- la sfasit voi goli fiserul cu solutia de pe Fenrir (input+output)

- programe C Linux

- voi stoca codul celui care adauga problema

- voi trimite prin SCP pe fenrir (cu contul celui care adauga problema) si voi compila codul

- voi prelua fisierul cu codul compilat si il voi stoca

- cand cineva ii rezolva problema, prin SSH voi trimite codul programului pe fenrir (cu contul celui care rezolva problema), il voi compila si il voi executa

- apoi voi trimite solutia corecta compilata si o voi executa

- citirea si afisarea se vor face cu fisiere

- voi compara cele 2 rezultate

- bool **ssh2\_scp\_send** ( resource $session , string $local\_file , string $remote\_file [, int $create\_mode = 0644 ] )

- bool **ssh2\_scp\_recv** ( resource $session , string $remote\_file , string $local\_file )

- la sfasit voi goli fiserul cu solutia de pe fenrir